Formal Concept Analysis Approximations via Atomic Priming

Ruairí de Fréin

Abstract Formal Concept Analysis (FCA) looks to decompose a matrix of objects-attributes into a set of sparse matrices capturing the underlying structure of a formal context. We propose a Rank Reduction (RR) method to prime approximate FCAs, namely RRFCA. While many existing FCA algorithms are complete, lectic ordering of the lattice may not minimize search/decomposition time. Initially, RRFCA decompositions are not unique or complete; however, a set of good closures with high support is learned quickly, and then, made complete. RRFCA has its novelty in that we propose a new multiplicative two-stage method. First, we describe the theoretical foundations underpinning our RR approach. Second, we provide a representative exemplar, showing how RRFCA can be implemented. Further experiments demonstrate that RRFCA methods are efficient, scalable and yield time-savings. We demonstrate the resulting methods lend themselves to parallelization.

Key words: Formal Concept Analysis, Rank Reduction, Factorization.

1 Introduction

Formal Concept Analysis (FCA) leverages the notion of a concept, an object-attribute building block of a binary relational dataset, and its ranking in a concept hierarchy to mine data-sets [25]. One short-coming is that concepts are mined according to lectic ordering and not concept *importance* or *support* in the formal context. Lectic ordering recommends itself on account of its thoroughness [10]. Mining times are typically long: this is demonstrated in [26], where the Twister Map-Reduce framework [8] is used to parallelize computational effort. However, in many cases

Ruairí de Fréin
Telecommunications Software & Systems Group, Waterford Institute of Technology, Ireland. rdefrein@gmail.com This is a draft version of the final article which appears on SpringerLink: http://dx.doi.org/10.1007/978-3-642-38317-5_6

some notion of Formal Concept (FC) importance might yield a better ordering, for example in knowledge discovery [17], information retrieval [21], and social networking analysis [22] applications. In this paper we leverage FC disjointness (as an embodiment of *importance*) along with lectic ordering to propose a new multiple-starting point approach, RRFCA, that improves mining and searching speed.

1.1 Related Work

Popular approaches for FCA include Ganter's algorithm [9], Lindig's algorithm [19], CloseByOne [15, 1] and their variants [24, 13]. The theoretical and empirical complexity of various approaches is compared by Kuznetsov in [16]. Computational complexity is the main measure for comparing algorithms: Kuznetsov and Obiedkov focus on the properties of the data ensemble, namely sparsity, the primary complexity inducing characteristic of the decomposition. Aside from sparsity, the main bottlenecks are memory and processing constraints. Ganter's algorithm computes concepts iteratively based on the previous concept, without incurring exponential memory requirements, by exploiting lectic ordering. CloseByOne produces many concepts in each iteration. Bordat's algorithm, described in [3], introduces a data structure to store previously found concepts, which results in considerable time-savings. This approach is made more efficient in [2] by removing the need for a structure of exponential size.

A significant short-coming of batch approaches is that the entire lattice must be reconstructed if the database changes. Incremental approaches have been made popular by Norris in [20], Dowling in [7], Godin et al. in [11], Carpineto and Romano in [4], Valtchev et al. in [23] and Yu et al. in [27] as they update the lattice structure when a new object is added to the database. To address the aforementioned memory and computational bottleneck, some parallel and distributed algorithms have been proposed. Krajca et al. proposed a parallel version based on CloseByOne in [13]. The first distributed algorithm [14] was developed by Krajca and Vychodil in 2009 using the Map-Reduce framework [6]. The authors of [26] proposed an efficient, distributed FCA implementation[1] using the Twister Map-Reduce framework [8]. Here, we look to the memory and computation challenge by using rank reduction method and *disjointness* to select good starting-intents for FCA.

The justification goes as follows: all concepts are not equal in a binary relational dataset. FC *support* (the extent to which it overlaps with the formal context) and its *expressiveness* (FC disjointness given a set of FCs), may be different for FCs. NextClosure's lectic ordering does not consider these concerns. To address this, we prime NextClosure with multiple starting-intents by taking reduced rank approximations of the binary relation using Nonnegative Matrix Factorization (NMF) [18]. Lectic ordering is then used from multiple starting points to generate the entire family of concepts, namely the Galois lattice, in a principled way. The underlying

[1] available at `https://github.com/TSSG/MRGanterPlus`

property of FCA –that concept intents are closed under intersection [10], namely closure– is leveraged to mine all closures as before.

1.2 Contributions

We propose an algorithm that learns a low rank factorization first and then extends it to an overcomplete representation.

This is a two-step algorithm that selects a subset (cardinality R) of entries from the formal context, from the entire set of entries (cardinality M), that minimizes representation error. Solving this problem by enumerating all possible choices in generally untractable. We relax the binary element-wise constraints on NMF's factors to make headway, solving a related convex optimization for each factor.

We make the link between basis selection problems, rank-1 approximations and closures.

NMF is not suited to learning overcomplete representations. Nonnegative sparse coding deploys regularization to address this [12]. Using FCA in tandem with NMF yields the complete family of FCs: RRFCA yields an overcomplete binary NMF (we borrow the idea of overcompleteness from frame theory). The cardinality of the NMF/RRFCA concept-set is easily augmented. Arranging binary-relational data as a concept lattice yields a powerful and intuitive representation of the dataset [25, 5]. Preservation of hierarchy is important as it facilitates a complete search. Even though we propose an approach where NextClosure starts from R different starting intents, a complete search is performed: the main characteristics of the FCA solution, completeness and hierarchy, are maintained. This paper is organized as follows. § 2 introduces the idea of an atom and relates it to the closure. § 3 defines the problem solved by NMF and shows how NextClosures makes NMF's representation overcomplete. The RRFCA algorithm and evaluation are presented in § 4 and § 5 respectively.

2 FCA: Taking an Atomic Point-of-View

We introduce the notational conventions used in the sequel to describe the formal context. Let O and P denote a finite set of objects and attributes respectively. In this paper, the data ensemble, S, is arranged in Boolean matrix form[2]. The related binary matrix is $S \in \mathbb{R}_{01}^{M \times N}$. There are M objects and N attributes. The objects and

[2] Boolean variables and matrices are used interchangeably with their corresponding binary values and matrices for simplicity.

Table 1 An example of a Formal Context (S). The value 1 in each row ($t \in O$) indicates that an object (t) has the corresponding attribute ($p \in P$). More generally, for a set of attributes $Y \subseteq P$, it is possible to identify the corresponding set of objects ($X \subseteq O$). All FCs are tabulated in Table 2.

	a	b	c	d	e	f	g
1	1	1	0	1	0	1	0
2	1	0	1	0	1	0	1
3	0	1	1	1	0	1	1
4	0	1	0	1	1	0	0
5	1	0	0	1	1	1	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
10	0	1	1	0	0	1	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

attributes are listed along the rows and columns of the matrix respectively; the value one is entered in a row-column position to denote that the object under consideration has that attribute; a zero entry denotes that this object does not have that attribute. Formally, this matrix describes the binary relation between the sets O and P; FCA looks to learn structures within this binary relation. The object X has attribute Y if $(X,Y) \in S$, $X \in O$ and $Y \in P$. The triple (O,P,S) is called a formal context. For example, in Table 1, $O = \{1,2,3,4,5,\ldots,10,\ldots\}$ and $P = \{a,b,c,d,e,f,g\}$, thus object $\{2\}$ has attributes $\{a,c,e,g\}$.

2.1 Underlying Association Structure: Closure

We define a derivation operator on X and Y where $X \subseteq O$ and $Y \subseteq P$ as a step towards generating FCA's association mechanism, namely the *closure*.

$$X' = \{p \in P \mid \forall t \in O : (t,p) \in S\}, \tag{1}$$

$$Y' = \{t \in O \mid \forall p \in P : (t,p) \in S\}. \tag{2}$$

The operation X' generates the set of attributes which are shared by all objects in X. Similarly, Y' generates the set of all objects which are common to all attributes in Y. A pair $\langle X,Y \rangle$ is called a FC of (O,P,S) if and only if $X \subseteq O$, $Y \subseteq P$, $X' = Y$, and $Y' = X$. Given a FC, $\langle X,Y \rangle$, X and Y are called its extent and intent. The crucial property of a FC is that the mappings $X \mapsto X''$ and $Y \mapsto Y''$, hereupon known as *closure operators*, hold. The closure operator can be used to calculate the extent and intent that form a FC; building blocks of the formal context are revealed by applying the closure mechanism methodically.

Establishing a sub/super-concept hierarchy allows for thorough, systematic FCA. Given $X_1, X_2 \subseteq O$ and $Y_1, Y_2 \subseteq P$ the concepts of a context are ordered as follows:

$$\langle X_1,Y_1 \rangle \leqslant \langle X_2,Y_2 \rangle :\Longleftrightarrow X_1 \subseteq X_2 \Longleftrightarrow Y_2 \subseteq Y_1 \tag{3}$$

an ordering which is interesting because it facilitates the iterative formation of a complete lattice which is called the concept lattice of the context [10].

We motivate, using Table 1, the disadvantage of choosing the starting-intent systematically (and blindly, e.g. without consideration of the support or disjointness of different structures in the formal context). NextClosure's concept lattice is anchored by the empty intent set $\{\}$. Consider the problem of selecting the best match for an arbitrary object with the objects in the relation matrix in Table 1. A cursory glance suggests that priming the search by comparing the object with rows $10, 4$ and 1 first, might give a good initial estimate for the best fit. However, determining that these rows are suitable is difficult. Starting a search methodically from an empty intent set $\{\}$ may incur more comparisons. Secondly, given the task of mining this binary-relation for all closures, a sensible approach would be to start out by considering these three starting-intents as they are *representative* of the dataset, and then to generate the closures related to them, using some scheme that minimizes redundant computation. We introduce some terminology to generalize the idea of a FC and arbitrary sub-structures in the formal context.

2.2 An Atomic Viewpoint

Definition 1 *An* atom *of a formal context is any matrix, F, formed from a non-empty subset $\hat{S} \in \mathscr{P}(S) \setminus \{\}$, of the non-zero entries of the binary matrix S.*

$$F_{t,p} = \begin{cases} 1, & (t,p) \in \hat{S} : \hat{S} \in \mathscr{P}(S) \setminus \{\} \\ 0, & \textit{otherwise.} \end{cases} \tag{4}$$

An atom is described by the set of pairwise indices, or a binary matrix. Set and matrix notation are used interchangeably. The powerset of S is $\mathscr{P}(S)$.

Ex. 1 *To fix ideas and notation, atoms in Table 1 include:*

- *the entire matrix S or the index set $\{\{1,1\},\{1,2\},\ldots\}$;*
- *row entries of the matrix $S_{1,:} = [1,1,0,1,0,1,0]$ or $\{\{1,1\},\{1,2\},\{1,4\},\{1,6\}\}$;*
- *proper subsets of row entries, $[1,0,0,1,0,1,0]$, or $\{\{1,1\},\{1,4\},\{1,6\}\}$;*
- *any proper subset of the matrix S, for example, $S_{1,1} = [1]$ or $\{1,1\}$.*

To relate these ideas to FCA, the closure, $\langle\{4,5\},\{d,e\}\rangle$ of S may also be expressed as an atom of S, $\{\{4,4\},\{4,5\},\{5,4\},\{5,5\}\}$: this notation is cumbersome however. Definition 1 is needed as the closure is too restrictive to describe all structure types in the dataset; most individual attribute and object sets have an associated closure, $\langle\{1,3,5,10\},\{f\}\rangle$, $\langle\{1,3,4,5\},\{d\}\rangle$ and $\langle\{5\},\{a,d,e,f\}\rangle$; the attribute g appears in the closures $\langle\{2,3,10\},\{c,g\}\rangle$, $\langle\{3,10\},\{b,c,f,g\}\rangle$, $\langle\{3\},\{b,c,d,f,g\}\rangle$, $\langle\{2\},\{a,c,e,g\}\rangle$, $\langle\{\},\{a,b,c,d,e,f,g\}\rangle$.

Property 1 *Formally, it is a property of closures that closures are atoms; however, atoms are not necessarily closures.*

Ex. 2 *The atom* $\{\{1,1\},\{1,2\},\{1,4\},\{1,6\},\{2,3\}\}$ *is not a closure: yet, the atom* $\{\{4,4\},\{4,5\},\{5,4\},\{5,5\}\}$ *is a closure. An atomic view of the formal context is useful (and cheaper) as it lends itself to an analysis of atom support, which can prime closure selection based on atomic importance via rank reduction.*

Property 2 *The closure associated with a row-atom is generated by applying the derivation operator twice.*

Ex. 3 *Given the atom* $[0,0,0,1,1,0,0]$ *which denotes a proper subset of the properties,* $Y = \{d,e\}$ *in Table 1, application of the derivation operator once yields* $Y' = \{4,5\}$*, and a second time yields,* $Y'' = \{d,e\}$*, which is a closure by definition.*

$$[0,0,0,1,1,0,0]'' \text{ or } \{d,e\}'', \quad \text{generates } \langle\{4,5\},\{d,e\}\rangle. \tag{5}$$

We overload the derivation (Eqn. 1) and closure operators to reduce notation.

Application of the closure operation on selected atoms –selected by hereto unspecified means– can be used to derive expressive closures whereas all closures must be computed to compare relative closure disjointness. Examining atoms is cheaper than examining closures as it does not involve complete closure computation up-front.

Property 3 *Closures are rank-1 approximations of the formal context. If* $X \subseteq O$*,* $Y \subseteq P$*,* $X' = Y$*, and* $Y' = X$*, let us construct the vectors, where each entry takes*

$$x_t = \begin{cases} 1, & \text{if } t \in X \\ 0, & \text{if } t \notin X, \end{cases}, \quad y_p = \begin{cases} 1, & \text{if } p \in Y \\ 0, & \text{if } p \notin Y, \end{cases}, \text{ then, } \operatorname{rank} xy^T = 1. \tag{6}$$

Ex. 4 *We demonstrate Property 3: the closure* $F_6 = \langle\{4,5\},\{d,e\}\rangle$ *is written as the outer product of two vectors,* $F_6 = xy^T$*, the vectors associated with the objects and attributes of the closure, a rank-1 approximation of S. This product is a special case of the Kronecker product of matrices.*

$$F_6 = \Big[\big[0,0,0,1,1,\ldots,0,\ldots\big]^T [0,0,0,1,1,0,0]\Big]^T = \begin{bmatrix} 0\,0\,0\,0\,0\ldots 0\ldots \\ 0\,0\,0\,0\,0\ldots 0\ldots \\ 0\,0\,0\,0\,0\ldots 0\ldots \\ 0\,0\,0\,1\,1\ldots 0\ldots \\ 0\,0\,0\,1\,1\ldots 0\ldots \\ 0\,0\,0\,0\,0\ldots 0\ldots \\ 0\,0\,0\,0\,0\ldots 0\ldots \end{bmatrix}^T \tag{7}$$

On examination of (Eqn. 7), the support of F_6 and accuracy of its approximation of S, may be measured by computing the overlap between the association matrix S and F_6: a count of the number of ones in F_6, measured using the degree of overlap of each FC F_i in matrix form F_i, with the formal context matrix S, $d_i = 1_M^T F_i 1_N$, for each FC, $F_i \in \mathscr{F}$, where $\mathscr{F}$ is the set of all FCs and 1_M^T is a vector of ones of size $1 \times M$. From Property 3, for formal contexts with rank greater than one, if we increase the number of closures in our approximation of the formal context in a judicious way, the quality of our approximation improves. The rate of improvement depends on the choice of each successive atom.

Proposition 1 *We posit that lectic ordering is a sub-optimal way to improve the choice of the next FC.*

Ex. 5 *A good choice of rank-*3 *FCs for S (with only one overlapping element) is the lead FC (LFC) set* $\mathscr{F}_l = \{F_5, F_{13}, F_{16}\}$ *(cf. Table 2 for the complete FC set).*

$$\hat{S} = \sum_{F_i \in \mathscr{F}_l} F_i = \begin{bmatrix} 0\,1\,0\,0\,1\ldots 0\ldots \\ 0\,0\,1\,0\,0\ldots 1\ldots \\ 0\,0\,1\,0\,0\ldots 1\ldots \\ 1\,0\,1\,0\,1\ldots 0\ldots \\ 0\,1\,0\,0\,1\ldots 0\ldots \\ 1\,0\,1\,0\,1\ldots 1\ldots \\ 0\,0\,1\,0\,0\ldots 1\ldots \end{bmatrix}^T \approx \begin{bmatrix} 1\,1\,0\,0\,1\ldots 0\ldots \\ 1\,0\,1\,1\,0\ldots 1\ldots \\ 0\,1\,1\,0\,0\ldots 1\ldots \\ 1\,0\,1\,1\,1\ldots 0\ldots \\ 0\,1\,0\,1\,1\ldots 0\ldots \\ 1\,0\,1\,0\,1\ldots 1\ldots \\ 0\,1\,1\,0\,0\ldots 1\ldots \end{bmatrix}^T \tag{8}$$

The fraction of ones in the rows of the approximation overlapping with S is $\{\frac{2}{3}, \frac{1}{2}, \frac{2}{3}, \frac{3}{4}, \frac{2}{3}, \frac{4}{4}, \frac{2}{3}\}$. *Adding the rank-1 matrix* F_{12} *to the set* $\mathscr{F}_l$ *improves the approximation without overlapping with any of the existing closures, adding three ones in the correct places. Adding the closure* F_6 *does not: this closure overlaps with existing closures in* $\mathscr{F}_l$. *Fig. 1 illustrates the order closures are mined by NextClosure. Using lectic ordering, closures* F_1-F_4 *have no overlap. However,* F_5 *overlaps with* 6 *of the previously accounted for elements (by* F_1-F_4*). Eleven of the formal context elements are accounted for by the LFC set* $\mathscr{F}_l = \{F_1, F_2, F_3, F_4, F_5\}$*, with six overlapping elements. In comparison, sixteen elements are accounted for by the LFC set*

$\mathscr{F}_l = \{F_5, F_{13}, F_{16}\}$ which is smaller, with one overlapping element. In conclusion, the LFC set $\mathscr{F}_l = \{F_5, F_{13}, F_{16}\}$ gives a better *representation with fewer FCs. In terms of searching through the formal context for association rules, given the intent $\{a, e\}$, navigating the arrangement of associations in Fig. 1 takes* 16 *comparisons, whereas the arrangement in Fig. 2 takes* 3 *comparisons with the members of the FC set.*

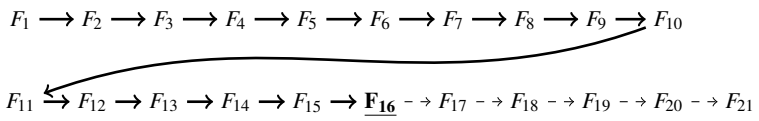

Fig. 1 The long way around: Given the initial intent $\{\}$ it takes 16 comparisons (indicated by full line-arrows) to find (or generate) the intent $\{a, e\}$ (underlined-bold).

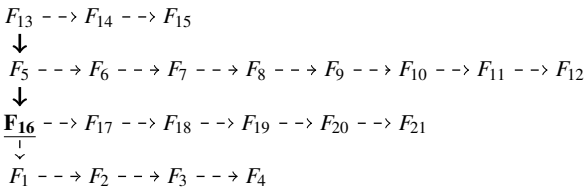

Fig. 2 Reduced rank speed-up: Given the initial set of lead intents it takes 3 comparisons (indicated by full line-arrows) to find (or generate) the intent $\{a, e\}$ (underlined-bold).

Problem Statement 1 *We desire the set, $\mathscr{F}_l$, of minimum cardinality $R \in \mathbb{Z}$ of closures, required to capture the structure of the matrix S with no overlap between closures, where $\odot$ denotes element-wise multiplication:*

$$\min R \quad \textit{subject to } ||S - \sum_{F_i \in \mathscr{F}_l} F_i||_2 = 0, \textit{ and } F_i \odot F_j = 0, \forall F_i, F_j \in \mathscr{F}_l. \tag{9}$$

Of secondary importance is the order in which closures should be generated in order to improve the coverage of the FC set of the formal context iteratively.

To answer the first question we start by examining an FCA of the formal context in Table 1, and define a measure of closure overlap. In order to make some progress, we then relax some of the constraints and examine atoms first. We reason that practicioners will want the ability to select R to optimize implementation. This simplifies the problem considerably.

Remark: Assessment of the expressiveness of each possible FC set requires computations of similarity and representativeness of all, and between all members of the

Table 2 FCs mined from a subset of relations in Table 1. Note that the empty concepts are included for completeness.

F_1: $\langle\{1,2,3,4,5,10\},\{\}\rangle$	F_8: $\langle\{1,3,4,10\},\{b\}\rangle$	F_{15}: $\langle\{1,2,5\},\{a\}\rangle$
F_2: $\langle\{1,3,5,10\},\{f\}\rangle$	F_9: $\langle\{1,3,10\},\{b,f\}\rangle$	F_{16}: $\langle\{2,5\},\{a,e\}\rangle$
F_3: $\langle\{2,4,5\},\{e\}\rangle$	F_{10}: $\langle\{1,3,4\},\{b,d\}\rangle$	F_{17}: $\langle\{1,5\},\{a,d,f\}\rangle$
F_4: $\langle\{1,3,4,5\},\{d\}\rangle$	F_{11}: $\langle\{1,3\},\{b,d,f\}\rangle$	F_{18}: $\langle\{5\},\{a,d,e,f\}\rangle$
F_5: $\langle\{1,3,5\},\{d,f\}\rangle$	F_{12}: $\langle\{4\},\{b,d,e\}\rangle$	F_{19}: $\langle\{2\},\{a,c,e,g\}\rangle$
F_6: $\langle\{4,5\},\{d,e\}\rangle$	F_{13}: $\langle\{3,10\},\{b,c,f,g\}\rangle$	F_{20}: $\langle\{1\},\{a,b,d,f\}\rangle$
F_7: $\langle\{2,3,10\},\{c,g\}\rangle$	F_{14}: $\langle\{3\},\{b,c,d,f,g\}\rangle$	F_{21}: $\langle\{\},\{a,b,c,d,e,f,g\}\rangle$

set, as we must not consider atoms in isolation –the list of all possible sets is long. Indeed this is a combinatorial optimization problem.

Definition 2 Closure Disjointness: *How well a closure represents a formal context is measured by 1) Target closure to Data Ratio (TDR) and 2) Target closure to Interfering closure Ratio (TIR). TDR gives the fraction of the formal context elements overlapping with the FC F_i. If we define an interferer atom to be $Y_i = \sum_{j\in\mathscr{F}\setminus F_i} F_i$, the sum of all closures other than F_i, Y_i is a non-binary matrix. Instead we use the element-wise union of these concepts: $Y_i = \bigcup_{j\in\mathscr{F}\setminus F_i} F_i$ which is binary.*

$$TDR_i = \frac{1_M^T F_i 1_N}{1_M^T S 1_N}, \quad TIR_i = \frac{1_M^T F_i 1_N}{1_M^T F_i \odot Y_i 1_N}, \tag{10}$$

where $\odot$ denotes element-wise multiplication. Closure disjointness is then defined as:

$$CD_i = TDR_i - \frac{TDR_i}{TIR_i} = \frac{1_M^T F_i 1_N - 1_M^T F_i Y_i 1_N}{1_M^T S 1_N} \tag{11}$$

When $F_i = S$, the formal context has rank-1 and $TDR_i = 1$. Typically, $0 \leq TDR_i \leq 1$. The interfering term $TIR_i = \infty$ when the FC overlap is zero, $F_i \odot Y_i = 0$. This measure of closure disjointness may be extended to testing for the disjointness of a set of closures from the rest, here we set $F_i = \cup_{j\in\mathscr{F}_l} F_j$, where $\mathscr{F}_l$ are members of the target set.

3 Atomic Decompositions for Formal Concept Analysis

We propose an algorithm that generates the starting-intents, like those in Table 2, by solving two convex optimization problems. Decomposing complicated multi-variate observations of some phenomenon into parts-based representations, representative atoms, yields an insight into the latent inner-workings of the process or model which

generated the data. The word parts-based is crucial here: we use parts-based decompositions as proxies for disjoint decompositions. Parts-based decompositions are typically well-placed, with respect to lectic ordering, to speed-up mining and searching routines. We use a heuristic to prime FCA: consider the following problem which is an intermediate step between Problem 1 and the NMF problem we solve.

Problem Statement 2 *Given a binary association matrix $S \in \mathbb{R}_{01}^{M\times N}$, for a given R,*

$$\begin{aligned} &\textit{minimize } ||S-OP||_2^2 \\ &\textit{subject to } O \in \mathbb{R}_{01}^{M\times R}, \quad P \in \mathbb{R}_{01}^{R\times N}, \\ &\qquad O_{:,j} \odot O_{:,k} = 0, \forall j,k, \quad P_{q,:} \odot P_{w,:} = 0, \forall q,w \end{aligned} \tag{12}$$

As a first step, we obtain a solution to Problem 2 for a given R. We also relax the element-wise binary constraints on O and P.

3.1 Preparing the Atomic Cookbook: Priming using NMF

Problem Statement 3 *Given the matrix S, NMF decomposes S into the product of two matrices, $O \in \mathbb{R}_{+}^{M\times R}$ and $P \in \mathbb{R}_{+}^{R\times N}$ where all matrices have exclusively non-negative elements.*

Definition 3 *NMF-Frobenius: $D_F(S||OP) = \frac{1}{2}\sum_{m,n}|S_{m,n} - [OP]_{m,n}|^2$.*
A suitable step-size parameter was proposed by Lee and Seung [18] which results in two alternating, multiplicative, gradient descent updating algorithms:

$$O \leftarrow O \odot SP^T \oslash OPP^T, \quad P \leftarrow P \odot O^T S \oslash O^T OP \tag{13}$$

where $\odot$ represents element-wise multiplication, and $\oslash$ is element-wise division.

NMF learns nonnegative factors and not binary factors. We use NMF to select suitable starting points for FCA by taking a nonlinear approximation of the resulting factorization. To the best of our knowledge this is the first time NMF has been applied to prime FCA. Appealing to Property 3, when the factors are element-wise binary entries, e.g. $O \in \mathbb{R}_{01}^{M\times R}$ and $P \in \mathbb{R}_{01}^{R\times N}$ each outer product, $O_{:,r}P_{r,:}$ is a closure if it supports formal context. When these closures are disjoint, we approximate S by a sum of the outer products $O_{:,r}P_{r,:}$.

Multiplicative updates are advantageous as the factors never become negative; therefore, projection into the positive orthant is not required. Alternating between the O and P updates implies that the optimization problem is not convex: the optimization is convex in either O or P, while the other factor is held fixed. Although the

solution is not unique, NMF's parts-based property means that it serves as a good starting point for FCA as the NMF decomposition gives intents and extents that are parts-based, e.g. approximately disjoint with good support.

An NMF can be learned using multiplicative or additive gradient descent, projected, exponentiated gradient methods or 2nd order Newton methods. A variety of costs such as the Kullback Leibler Divergence and members of various divergence families have been used. Exploiting the sparse nature of the matrix S may yield more suitable solutions, for example when it is known that S has high sparsity; we focus on the traditional NMF approach (Eqn. 3),

$$D_F^{\alpha,\beta}(S||OP) = D_F(S||OP) + \alpha J_\alpha(O) + \beta J_\beta(P), \tag{14}$$

where $\alpha \geq 0$ and $\beta \geq 0$ are regularization parameters and the functions J_α and J_β enforce constraints on the factors (for example, sparsity or disjointness/orthogonality of columns or rows of the factors). FCA priming via NMF is best illustrated by example.

Ex. 6 Reduced-Rank Formal Concept Analysis (RRFCA): *Application of NMF to the subset of rows of S given in Table 1, with $R = 4$, and taking the nonlinear approximation of the resulting factors yields*

$$O_{m,r} \leftarrow \begin{cases} 1 & \text{if } O_{m,r} > .5 \\ 0 & \text{otherwise} \end{cases} \quad P_{r,n} \leftarrow \begin{cases} 1 & \text{if } P_{r,n} > .5 \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

and thus the factorization (for an abridged version of the entries in Table 1):

$$S \approx OP = \begin{bmatrix} 1&0&0&0&1&0 \\ 0&0&1&0&0&1 \\ 0&0&1&1&0&0 \\ 0&1&0&0&0&0 \end{bmatrix}^T \begin{bmatrix} 0&0&0&1&0&1&0 \\ 0&1&1&0&0&1&1 \\ 0&0&0&0&0&0&0 \\ 1&0&0&0&1&0&0 \end{bmatrix} \tag{16}$$

Looking to the matrix P, because $M > N > R$ in this case, we use the intents $\{d, f\}$, $\{b, c, f, g\}$ and $\{a, e\}$ to generate the closures F_5, F_{13} and F_{16}, using the closure operator on each intent in turn[3]*. For completeness, it is possible to check the rank-*1 *estimate, $O_{:,r}P_{:,r}$ is a closure by verifying $S - O_{:,r}P_{:,r} > 0$. Table 3 illustrates the location of the FCs yielded by NMF relative to those mined by lectic ordering. Notice the intents generated are approximately evenly distributed through the table. We do not include the empty intent set generated by the factorization as this is an artefact of the non-linear approximation's parametrization.*

[3] The choice of operating on intents or extents is based on the dimensions M and N.

Table 3 Distribution of the intents in the concept lattice: Underlined intents denote the intents learned by NMF which are used as starting-intents for FCA. Three intents are mined from F_1 (F_2–F_4), seven intents are mined from F_5 (F_6–F_{12}), two intents are mined from F_{13} (F_{14}–F_{15}), and finally, five are mined from F_{16} (F_{17}–F_{21}).

F_1: $\langle\{1,2,3,4,5,10\},\{\}\rangle$	F_8: $\langle\{1,3,4,10\},\{b\}\rangle$	F_{15}: $\langle\{1,2,5\},\{a\}\rangle$
F_2: $\langle\{1,3,5,10\},\{f\}\rangle$	F_9: $\langle\{1,3,10\},\{b,f\}\rangle$	$\underline{F_{16}}$: $\underline{\langle\{2,5\},\{a,e\}\rangle}$
F_3: $\langle\{2,4,5\},\{e\}\rangle$	F_{10}: $\langle\{1,3,4\},\{b,d\}\rangle$	$\overline{F_{17}}$: $\overline{\langle\{1,5\},\{a,d,f\}\rangle}$
F_4: $\langle\{1,3,4,5\},\{d\}\rangle$	F_{11}: $\langle\{1,3\},\{b,d,f\}\rangle$	F_{18}: $\langle\{5\},\{a,d,e,f\}\rangle$
$\underline{F_5}$: $\underline{\langle\{1,3,5\},\{d,f\}\rangle}$	F_{12}: $\langle\{4\},\{b,d,e\}\rangle$	F_{19}: $\langle\{2\},\{a,c,e,g\}\rangle$
$\overline{F_6}$: $\overline{\langle\{4,5\},\{d,e\}\rangle}$	$\underline{F_{13}}$: $\underline{\langle\{3,10\},\{b,c,f,g\}\rangle}$	F_{20}: $\langle\{1\},\{a,b,d,f\}\rangle$
F_7: $\langle\{2,3,10\},\{c,g\}\rangle$	$\overline{F_{14}}$: $\overline{\langle\{3\},\{b,c,d,f,g\}\rangle}$	F_{21}: $\langle\{\},\{a,b,c,d,e,f,g\}\rangle$

3.2 RRFCA: Mining for overcompleteness using NextClosure

Given the starting-intents in Ex. 6, closures are generated iteratively using lectic ordering which is defined *ab initio* by RRFCA, by arranging P in an arbitrary linear order $p_1 < p_2 < \cdots < p_i < \ldots < p_N$. Appealingly, once lectic ordering is in place, closures are only generated once. We demonstrate the ordering, given two subsets $Y_1, Y_2 \subseteq P$. Y_1 is lectically smaller than Y_2 if the smallest element in which Y_1 and Y_2 differ belongs to Y_2.

Property 4 *A rank-*1 *formal context approximation is valid if it is a closure and it is lectically smaller than any FC already mined. Formally, the smallest element in which Y_1, Y_2 differ is p_i*

$$Y_1 \leq_{p_i} Y_2 :\Longleftrightarrow \exists_{p_i}(p_i \in Y_2, p_i \notin Y_1, \forall_{p_j<p_i}(p_j \in Y_1 \Longleftrightarrow p_j \in Y_2)). \quad (17)$$

Given the current intent, RRFCA uses (Eqn. 17) as a feasibility condition for accepting a new candidate FC like NextClosure.

Property 5 *Given a lectically ordered set P and three FCs, $F_i = \langle X_i, Y_i\rangle$, $F_j = \langle X_j, Y_j\rangle$, and $F_k = \langle X_k, Y_k\rangle$ from the family of concepts $\mathscr{F}$, where $Y_i < Y_j < Y_k$, we can mine all concepts that lie, lectically, between F_i and F_j, by starting at F_i and stopping when the FC is F_j, and all concepts between F_j and F_k. As the FCs F_i, F_j, F_k are valid closures (rank-*1 *approximations of S), the feasiblity condition ensures that only lectically smaller intents will be generated in each range. We may mine in parallel using many lectically ordered start-stop pairs.*

Property 5 arises from the combination of Properties 4 and 3. All that remains is to formulate an algorithm that selects good starting intents: intents that are well-spaced lectically. We use parts-based intents –NMF is a good generation procedure.

Ex. 7 *Looking to Table 3, the order of the attributes P, is defined as* $a < b < c < d < e < f < g$. *Given a set of starting intents, the ordering* $\{\} < \{d,f\} < \{b,c,f,g\} < \{a,e\}$ *of the intents holds. Three intents are mined from* F_1 *(*F_2–F_4*), seven intents are mined from* F_5 *(*F_6–F_{12}*), two intents are mined from* F_{13} *(*F_{14}–F_{15}*), and finally, five are mined from* F_{16} *(*F_{17}–F_{21}*).*

To generate all closures systematically using lectic ordering, we use NextClosure's $\oplus$-operation starting from each intent in the LFC set generated by NMF:

$$Y \oplus p_i := ((Y \cap \{p_1, \ldots, p_{i-1}\}) \cup \{p_i\})'', \quad \text{where } Y \subseteq P \text{ and } p_i \subset P. \tag{18}$$

A new candidate FC is compared with the previous concept and the next lectically smaller LFC's attribute. If the condition in (Eqn. 17) is satisfied and the candidate is lectically larger than the next LFC, the candidate concept produced by (Eqn. 18) is kept and added to the lattice.

Algorithm 1 Rank Reduction Formal Concept Analysis (RRFCA)

Input: S, R, T: Binary relation matrix, rank of NMF, and nonlinear threshold T.
Output: $\mathscr{F}_l, \mathscr{F}$: LFCs set, and complete family of FCs.

1: Initialization: $S = S + eps$, Set $O \sim \mathscr{U}(0,1)$ and $P \sim \mathscr{U}(0,1)$. Ensure nonnegativity by adding a small constant, eps, to S, O and P. Select the lectic order P.
2: $[O,P] = \text{NMF}(S,O,P,R)$: Run NMF. Take the nonlinear approximation in (Eqn. 15) using T. Generate the set of lectically ordered LFCs $\mathscr{F}_l$. Add the empty intent set $\mathscr{F}_l = \langle O, \{\}\rangle \cup \mathscr{F}_l$, and $\mathscr{F}_l = \langle \{\}, P\rangle \cup \mathscr{F}_l$.
3: **if** $M > N$ and $|\mathscr{F}_l| = R + 1$ **then**
4: Determine R starting and stopping-intent pairs using $R+1$ valid distinct intents.
5: Call $R+1$ AllClosure processes passing successively lectically ordered starting/stopping-intents to each, $\{F_i, F_j\}, \forall i, j$ valid pairs.
$[\mathscr{F}_r] = \text{AllClosure}(F_i, F_j, S, O, P)$.
6: **end if**
7: Form the union of all process results $\mathscr{F} = \bigcup_r \mathscr{F}_r$. **return** $\mathscr{F}$

Algorithm 2 AllClosure - NextClosure Process r

Input: $\{F_i, F_j\}, S, O, P$: starting/stopping FCs.
Output: $\mathscr{F}_r$: FCs derived by process r.

1: Initiate process: $Y \leftarrow Y_i$.
2: **while** Y is not the last closure Y_j **do**
3: $[Y] = \text{NextClosure}(O, P, S, Y)$;
4: $\mathscr{F}_r \leftarrow \mathscr{F}_r \cup Y$;
5: **end while**
6: **return** $\mathscr{F}_r$

Algorithm 3 NextClosure – Process r

Input: O, P, S, Y: formal context & current intent.
Output: Y.

```
for p_i from p_N down to p_1 do
    if p_i ∉ Y then
        candidate ← Y ⊕ p_i;
        if candidate ≤_{p_i} Y then
            Y ← candidate;
            break;
        end if
    end if
end for
return Y
```

4 A Rank Reduced FCA Algorithm: RRFCA

Algorithm 1 takes as input the entire binary relation matrix S, or the matrix S formed from a uniform sampling or the row atoms of the entire association matrix to reduce complexity, where the size of the sampled matrix is $M' \times N$, and $M' < M$. We denote the whole matrix and some subset of the row-entries by S for simplicity. Alternatively, the input S may denote some partition of the entire dataset by extending the distributed FCA method proposed in [26]. Algorithm 1 initializes $\approx R+1$ NextClosure-like processes, described in Algorithms 2, 3, to mine all of the FCs in a given range F_i-F_j which allows for time-savings in FCA due to distribution. Each process returns a FC set $\mathscr{F}_r$ which is based on the set of FCs mined from the intents or extents learned by NMF, $\mathscr{F}_l$. In Fig. 2 the LFC set is $\mathscr{F}_l = \{F_1, F_5, F_{13}, F_{16}, F_{21}\}$. The pairs of starting-stopping intents generated by the nonlinearity of NMF are $\{Y_1, Y_5\}, \{Y_5, Y_{13}\}, \{Y_{13}, Y_{16}\}, \{Y_{16}, Y_{21}\}$. When the one-entries in the binary association matrix are uniformly distributed, as a rule-of-thumb, we note that intents with fewer elements generate more FCs than intents with more entries.

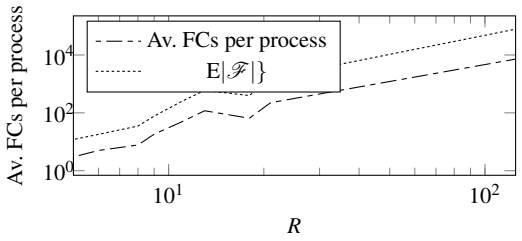

Fig. 3 Computation is divided by ≈ 5: The expected number of FCs mined per process is plotted against the rank of NMF in Fig. 4. Plotting the total number o f FCs mined illustrates the extent to which workload is distributed.

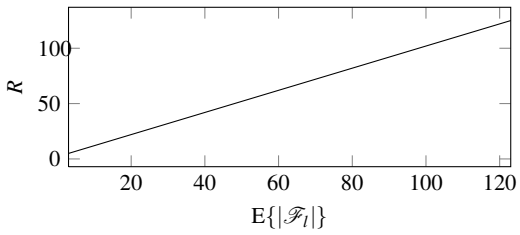

Fig. 4 Most rank-1 approx. yield FCs: Fig. 4 illustrates that priming via NMF is successful: the expected number of starting-stopping pairs is plotted against the rank of NMF. Almost all NMF factors yield a corresponding closure.

5 Empirical Evaluation

In this empirical evaluation we establish the properties of the RRFCA algorithm.

Proposition 2 *NMF parts-based representations generate LFCs which are disjoint and evenly distributed throughout the concept lattice.*

To illustrate this property, we compute the average number of FCs computed for each starting-stopping-intent pair, for a large ensemble of large random binary matrices. This value, namely the *lectic length*, of each LFC should be approximately equal for all FCs, so that workload is shared fairly. In traditional FCA, the lectic length equals the total number of FCs to be mined. We generate binary matrices of size $(10^i \times 5^i)$ for the powers $i = 1, 1.1, 1.2, \ldots$ We construct 20 random binary matrices of each dimension $(10^i \times 5^i)$ by drawning element-wise values from a Bernoulli distribution, where the probability of one is 0.2. Fig. 4 shows the expected lectic length against the NMF rank parameter R, for RRFCA and FCA. The expected lectic length is an indication of the number of FCs mined by each process, once the LFC set has been determined. The rank of NMF is set to grow according to $R = 3^i$, where i increases monotonically. For completeness, the expected number of starting-stopping pairs is also plotted against the rank of NMF, R in Fig. 4.

1) **Fig. 4 demonstrates that the average number of FCs learned per process is significantly smaller than the total number of FCs.** This is because the average workload is distributed across the number of processes plotted in Fig. 4, approximately $R+1$. All FCs are computed, yet each process mines only $\approx 20\%$ of the FCs. The duration of RRFCA runtime is reduced accordingly. It may be reduced further by increasing R.

2) **Fig. 4 demonstrates that a parts-based initialization procedure efficiently selects FCs.** The rank, R, is approximately equal to the expected number of starting-stopping FCs: the relationship is linearity. Here, all rank-1 approximations generate valid closures. Note that the number of LFC is augmented by two as the empty intent and full intent sets are concatenated to $\mathscr{F}_l$.

3) Searching for a target intent is performed by leveraging the fact that FCs are ordered lectically. A target intent is compared with the intent of each of the ordered LFCs $\mathscr{F}_l$, in order to find the appropriate intent range for the target intent, i.e. an upper and lower bound on the search range, namely the starting-stopping intent. Once the starting-stopping intent pair has been found, the target intent is found by

searching the FCs mined in that intent range. Fig. 2 illustrates this process. The intent of F_{19} is found by first determining that it lies in the range F_{16} and F_{21}, and then by searching within this range. The average lectic length plotted in Fig. 4 gives the average maximum search length on the number of comparisons required to locate a FC. **Organizing the search using LFCs reduces the expected maximum possible search length considerably (by a factor of 5 here) in comparison with NextClosure.**

Proposition 3 *RRFCA is scalable because the expected number of FCs learned per process is a linear function of the reciprocal of the expected number of starting-stopping pairs, which is bounded by the rank of the formal context.*

In conclusion, increasing R increases the number of starting-stopping intent pairs, which increases the amount of parallelization possible for the NextClosure mining step. It follows that the searching time and mining time is reduced. This incurs a higher computational cost on the NMF priming step. The complexity of the O and P NMF updates is given in flops as $2R\left[[MN+N]+2R^2\left[N+M\right]\right]$ flops per iteration. Recall that the purpose of the NMF step is not to find a good binary fit, but to find a rough initialization for FCA, when the element-wise values exceed $T > 0.5$. Thus, NMF may be run for far fewer iterations in this case, reducing the cost. In addition, we have experimented with uniformly sampling the set of objects to hot-start NMF-priming to good effect. To underline the utility of *parts-based* starting-stopping

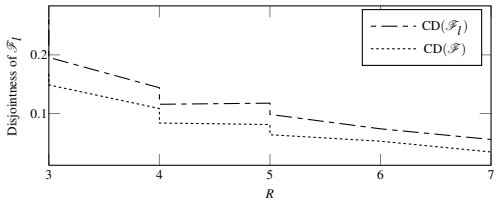

Fig. 5 The mean closure disjointness of the LFC set $\mathscr{F}_l$ is significantly higher than the mean closure disjointness of the total FC set $\mathscr{F}$. The LFC set generated using NMF priming gives a better representation of the formal context.

pairs, we plot the mean concept disjointness (Eqn. 11) of the starting-stopping pairs in Fig. 5. The idea is to demonstrate that the set $\mathscr{F}_l$ gives a good representation of the formal context, indeed, better than the full set of FCs. The mean disjointness of all members of the LFC set, computed using (Eqn. 11), is compared with the mean disjointness of the entire set of FCs. Recall: a good parts-based representation is one that has good support of the formal context and minimal overlap with the other members of the representation. Fig. 5 shows that the set $\mathscr{F}_l$ gives a good representation of the formal context without significant overlap. Disjointness of the LFC set is why the lead intents tend to be well distributed throughout the set of ordered FCs, and thus why mining can be distributed and search speeds be improved. As the number of LFCs is increased, the average disjointness is decreased; The disjointness characteristic yeilds a good ordering on how FCs are selected.

Primer Parametrization RRFCA relies on appropriate selection of the parameters R and the threshold $T = 0.5$; The accuracy of the factorization is only a sec-

ondary concern compared to learning good starting intents/extents. In this setting, the parameter R is the desired number of starting points for RRFCA. The only requirement[4] is that $R < M < N$. A larger choice of R incurs higher computational cost, and typically intents/extents of smaller cardinality. A smaller value of R yields intents/extents of larger cardinality. The values in the intent matrix P may be interpreted as the probability that a member of the intent set should be grouped with other intents in that set. Given this interpretation of a probability, the choice of $T = .5$ is justified: values of the factors are typically in the range $(0,1)$, and therefore the threshold $T = .5$ corresponds to .5 probability. Due to the non-uniqueness (and non-convexity) of the NMF decomposition, the set of starting point intents depends on the initialization of the factors O and P. They were initialized randomly from a uniform distribution $\mathscr{U}(0,1)$ over the range of values $(0,1)$ here. A small constant is added to the binary relation matrix and the factors to ensure nonnegativity. The success of the decompositions supports the notion that a range of parametrizations of the algorithm is permissible.

Overcompleteness Leveraging NMF as a primer for FCA poses the question of why we need FCA in the first instance. The answer is as follows: NMF is unsuited to binary data. The factors returned by NMF are real-valued, and applying a nonlinearity to the factors reduces the accuracy of the decomposition. However NMF is suited to giving a good starting point to another binary factorization technique, FCA. The strength of FCA lies in the ability to mine all FCs from the formal context. NMF is suited to learning low-rank approximations. Once an overcomplete representation of the type mined by FCA is required, NMF struggles. Attempts to regularize NMF factorizations using sparsity constraints on one factor (not reported here) have led to mixed results. Typically one factor is made sparse, yet the other is made more dense to compensate and improve the accuracy of the decomposition. FCA on the other hand is guaranteed to find all factors without side-constraints (Eqn. 14): FCA mines overcomplete representations. If the rank parameter R is set to be greater than M or N, NMF's accuracy will improve, but the intents and extents learned will not be as distinct as those of FCA.

Future Work

The equitable distribution of FCs analysis using RRFCA may be further improved. Earlier (lectically smaller) starting-stopping pairs learn fewer FCs than later (lectically larger) starting-stopping pairs due to the disjointness of the starting intents. In future work, the performance of RRFCA may be optimized by clustering ordered intent sets with fewer entries together so that the computational demands made of each process are evenly distributed. We have used the number of FCs computed by each process, e.g. the number of FCs computed between each starting-stopping-intent pair, to demonstrate the scalability of the approach, and not the total simulation run-time. The computation time of each closure computation depends on

[4] If we desire starting-extents.

the sparsity of the dataset, the speed of the various computational components of the implementation and platform, and the sparsity of the starting-stopping intents. We discuss our results in terms of the number of closures computed by each of our processes for a given level of dataset sparsity. In future work we will focus on platform/implementation specific optimizations.

6 Conclusions

The complexity associated with NextClosure's ordered approach incurs high computational expense, and bounds the size of the largest dataset NextClosure can practically process. This complexity is the main bottleneck of FCA. In this paper we introduced RRFCA, which exploits the fact that rank-1 approximations are closures, and tha lectic ordering of a set of representative closures can be used to sub-divide mining tasks. LFC are determined by solving two alternating convex optimization problems, which are a relaxed version of the original problem. RRFCA starts NextClosure from $\approx R+1$ different starting intents, allowing for parallelization of the mining process. Empirical evaluation demonstrates that mining speed-up of $R^{-1} * 100\%$ is achievable. This result is qualified by conditions on the rank of the formal context.

Acknowledgements This work was supported by grant 08 / SRC / I1403 FAME SRC and 11 / TIDA / I2024.

References

1. Andrews, S.: In-Close, a Fast Algorithm for Computing Formal Concepts. In: 17th Int. Conf. on Conceptual Structures (2009)
2. Berry, A., Bordat, J.P., Sigayret, A.: A Local Approach to Concept Generation. Annals of Mathematics and Artificial Intelligence **49**(1), 117–36 (2006)
3. Bordat, J.P.: Calcul pratique du treillis de Galois d'une correspondance. Mathématiques et Sciences Humaines **96**, 31–47 (1986). URL `http://www.zentralblatt-math.org/zmath/en/search/?q=an:0626.06007`
4. Carpineto, C., Romano, G.: A Lattice Conceptual Clustering System and Its Application to Browsing Retrieval. Machine Learning pp. 95–122 (1996)
5. Caspard, N., Monjardet, B.: The Lattices of Closure Systems, Closure Operators, and Implicational Systems on a Finite Set: A Survey. Disc. Appl. Math. pp. 241–69 (2003)
6. Dean, J., Ghemawat, S.: MapReduce: Simplified Data Processing on Large Clusters. Commun. ACM **51**(1), 107–13 (2008)
7. Dowling, C.E.: On the Irredundant Generation of Knowledge Spaces. J. Math. Psychol. **37**, 49–62 (1993). DOI 10.1006/jmps.1993.1003. URL `http://portal.acm.org/citation.cfm?id=162975.162978`
8. Ekanayake, J., Li, H., Zhang, B., Gunarathne, T., Bae, S.H., Qiu, J., Fox, G.: Twister: a Runtime for Iterative MapReduce. In: S. Hariri, K. Keahey (eds.) HPDC, pp. 810–8. ACM (2010). URL `http://dblp.uni-trier.de/db/conf/hpdc/hpdc2010.html#EkanayakeLZGBQF10`

9. Ganter, B.: Two Basic Algorithms in Concept Analysis. Formal Concept Analysis pp. 312–40 (1984)
10. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, Berlin-Heidelberg (1999)
11. Godin, R., Missaoui, R., Alaoui, H.: Incremental Concept Formation Algorithms Based on Galois (Concept) Lattices. Computational Intelligence **11**, 246–67 (1995). URL `http://dblp.uni-trier.de/db/journals/ci/ci11.html#GodinMA95`
12. Hoyer, P.O.: Non-negative sparse coding. In: Proc. IEEE Workshop on Neural Networks for Signal Processing, pp. 557–65 (2002)
13. Krajca, P., Outrata, J., Vychodil, V.: Parallel Recursive Algorithm for FCA. In: CLA 2008, vol. 433, pp. 71–82 (2008)
14. Krajca, P., Vychodil, V.: Distributed Algorithm for Computing Formal Concepts Using Map-Reduce Framework. In: 8th Int. Symp. on Intelligent Data Analysis, pp. 333–44. Springer-Verlag, Berlin, Heidelberg (2009). DOI http://dx.doi.org/10.1007/978-3-642-03915-7_29
15. Kuznetsov, S.O.: A Fast Algorithm for Computing All Intersections of Objects in a Finite Semi-Lattice. Automatic Documentation and Mathematical Linguistics **27(5)**, 11–21 (1993)
16. Kuznetsov, S.O., Obiedkov, S.A.: Comparing Performance of Algorithms for Generating Concept Lattices. J. of Exper. and Th. Artif. Intell. **14**, 189–216 (2002). URL `citeseer.ist.psu.edu/666686.html`
17. Lakhal, L., Stumme, G.: Efficient Mining of Association Rules Based on Formal Concept Analysis. In: Formal Concept Analysis'05, pp. 180–95 (2005)
18. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: NIPS, pp. 556–62. MIT Press (2000)
19. Lindig, C.: Fast Concept Analysis. Working with Conceptual Structures-Contributions to ICCS **2000**, 235–48 (2000)
20. Norris, E.M.: An Algorithm for Computing the Maximal Rectangles in a Binary Relation. Rev. Roum. Math. Pures et Appl. **23**(2), 243–50 (1978)
21. Polaillon, G., Aufaure, M.A., Grand, B.L., Soto, M.: FCA for Contextual Semantic Navigation and Information Retrieval in Heterogeneous Information Systems. In: DEXA Workshop, pp. 534–9 (2007)
22. Snásel, V., Horak, Z., Kocibova, J., Abraham, A.: Analyzing Social Networks Using FCA: Complexity Aspects. In: Web Intelligence/IAT Workshops'09, pp. 38–41 (2009)
23. Valtchev, P., Missaoui, R., Lebrun, P.: A Partition-based Approach Towards Constructing Galois (concept) Lattices. Discrete Mathematics pp. 801–29 (2002)
24. Vychodil, V.: A New Algorithm for Computing Formal Concepts. Cybernetics and Systems pp. 15–21 (2008)
25. Wille, R.: Restructuring Lattice Theory: an Approach Based on Hierarchies of Concepts. In: I. Rival (ed.) Ordered sets, pp. 445–70. Reidel (1982)
26. Xu, B., de Fréin, R., Robson, E., Ó Foghlú, M.: Distributed Formal Concept Analysis Algorithms Based on an Iterative MapReduce Framework. In: F. Domenach, D. Ignatov, J. Poelmans (eds.) FCA, *LNCS*, vol. 7278, pp. 292–308 (2012)
27. Yu, Y., Qian, X., Zhong, F., Li, X.R.: An Improved Incremental Algorithm for Constructing Concept Lattices. Software Engineering, World Congress **4**, 401–5 (2009). DOI http://doi.ieeecomputersociety.org/10.1109/WCSE.2009.301